


rasdaman
raster data management

JACOBS
UNIVERSITY

No One Size Fits All:
Accepting Datatype V)ariety

ISO/IEC JTC1 SC32 Big Data Analytics Study Group

Amsterdam / NL, 2014-05-15

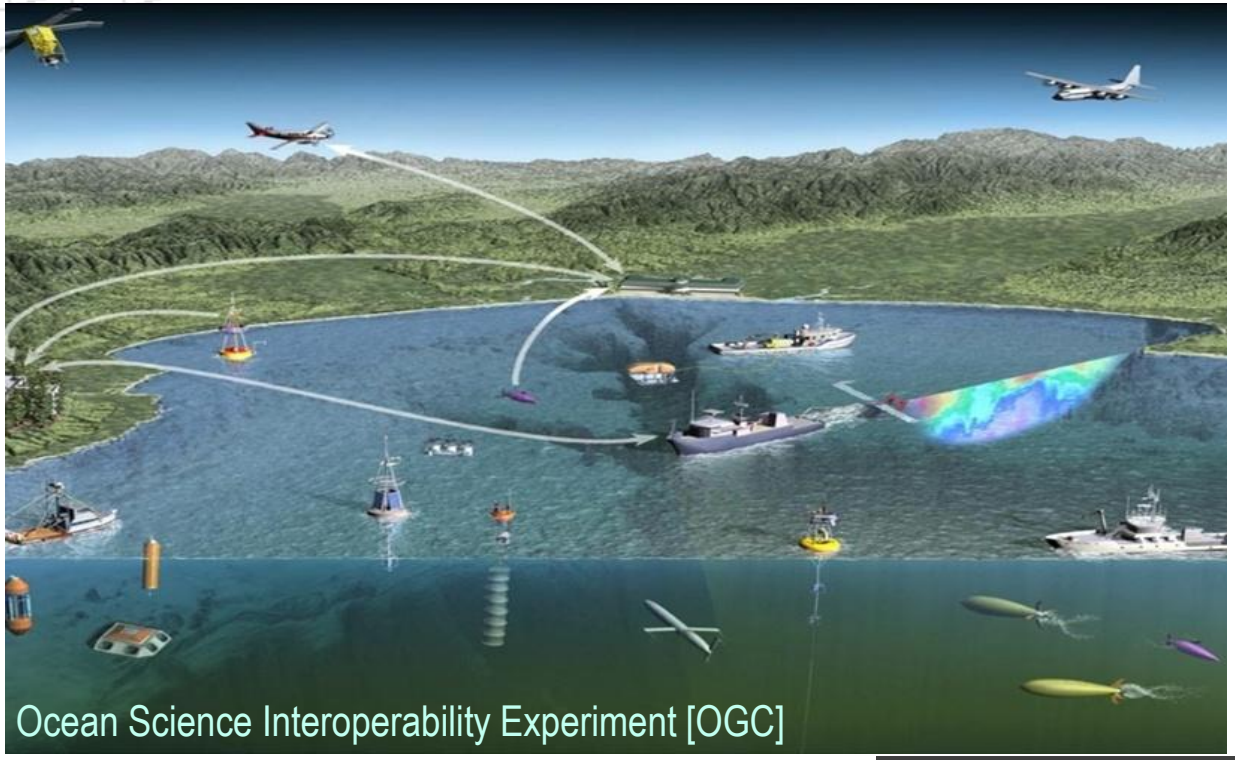
Peter Baumann

Jacobs University | rasdaman GmbH

p.baumann@jacobs-university.de

Big Data (not only) in Geo

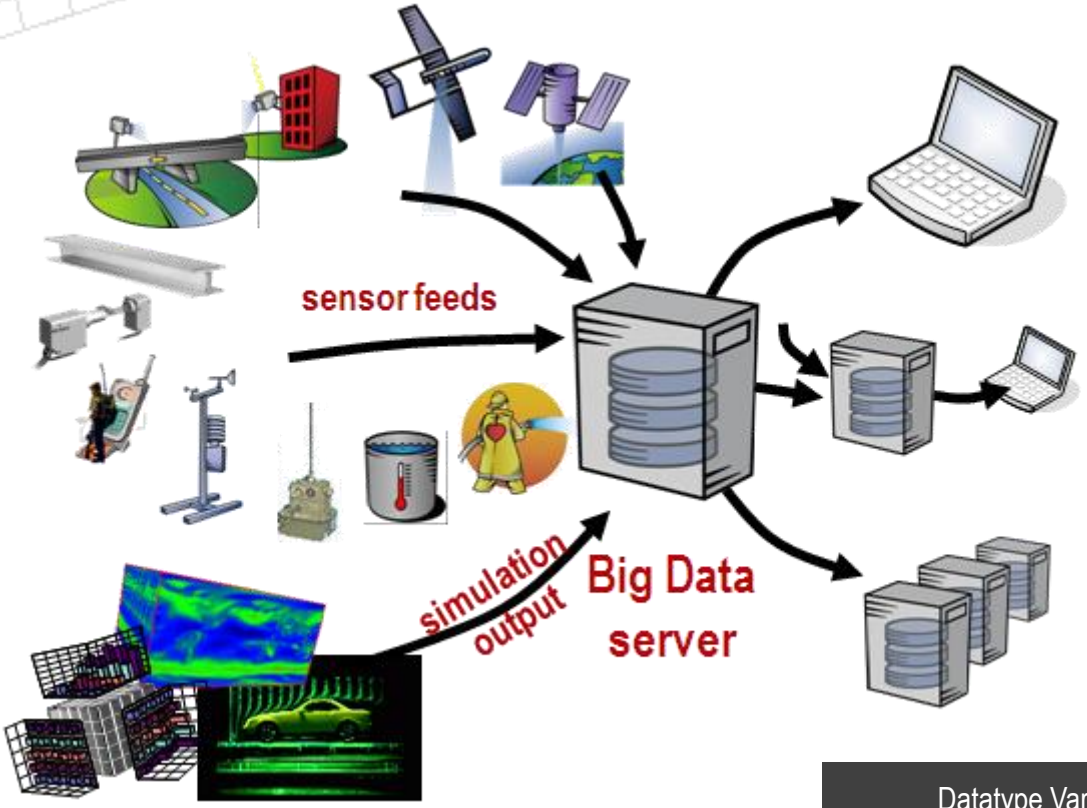
- Roughly: Spatio-temporal sensor, image, simulation, statistics data
  - Often n-D raster data = arrays
- „massive arrays & graphs next great challenges“ [major DBMS vendors]
  - „n-D array support critical to Big Data” [US SQL committee]



Ocean Science Interoperability Experiment [OGC]

Big Data (not only) in Geo

- Roughly: Spatio-temporal sensor, image, simulation, statistics data
  - Often n-D raster data = arrays
- „massive arrays & graphs next great challenges“ [major DBMS vendors]
  - „n-D array support critical to Big Data” [US SQL committee]



RM-ODP for Tackling Big Data

- OGC BigData Working Group utilizes RM-ODP
  - Reference Model for Open Distributed Processing
  - collaborative writing by variety of stakeholders
- Accommodates different perspectives:
  - **enterprise viewpoint:** purpose, scope and policies
  - **information viewpoint:** semantics of information & processing
  - **computational viewpoint:** functional decomposition, interfaces
  - **engineering viewpoint:** distribution of processing
  - **technology viewpoint:** choice of technology

Variety

- RM-ODP informational / computational viewpoint:
  at first glance, many different data structures:
  - Stock trading: 1-D sequences
  - Social networks: large, homogeneous graphs
  - Ontologies: small, heterogeneous graphs
  - Climate modelling: 4D/5D arrays
  - Satellite imagery: 2D/3D arrays (+irregularity)
  - Genome: long strings
  - Particular physics: sets of events
  - XML data: hierarchies
  - Key/value stores: sets of unique identifiers + whatever
  - etc.
- reducible to a few core structures: sets/bags; n-D arrays; graphs; trees; ...

Use Case 1: Arrays in SQL

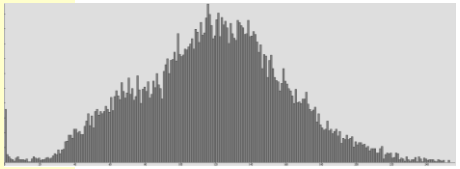
under discussion in ISO

- Arrays in DDL (extending SQL:1999 1-D arrays):

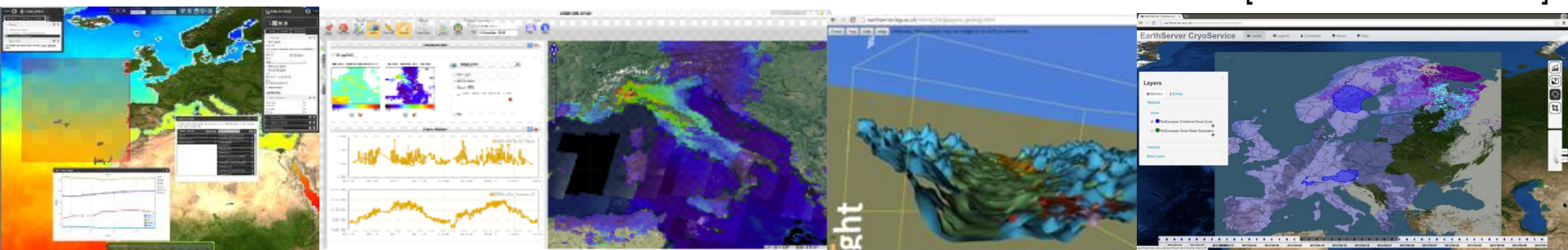
```
create table LandsatScenes(
    id: integer not null, acquired: date,
    scene: row( red: integer, ..., blue: integer ) array [ 0:4999,0:4999] )
```

- Arrays in DML:
  „band 1 histogram, in CSV, of Landsat scenes acquired in June 1990“

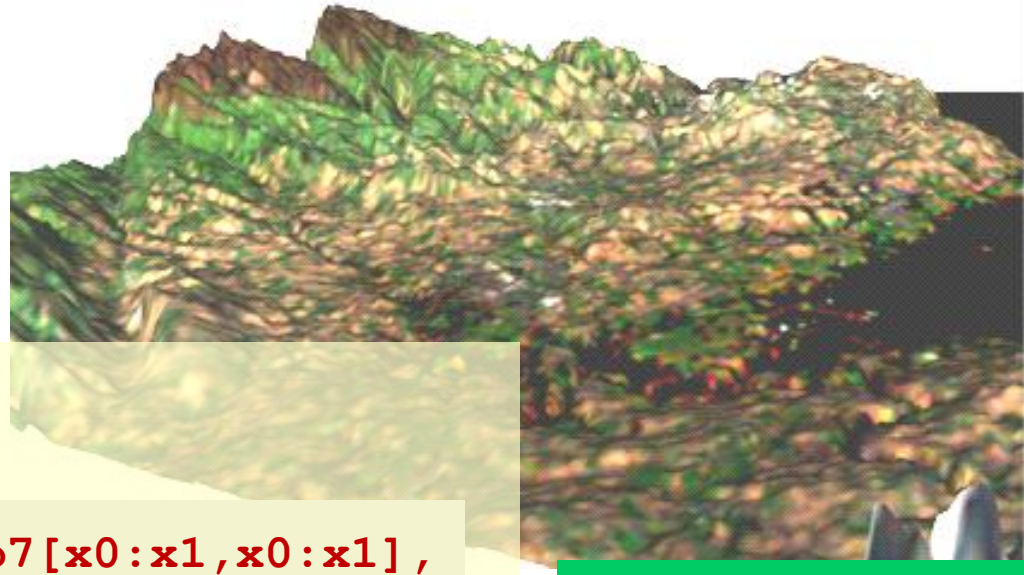
```
select encode(
            array b in [0:255]
            values b = count_cells( scene.band1 ),
         „csv“ )
from LandsatScenes
where acquired between „1990-06-01“ and „1990-06-30“
```



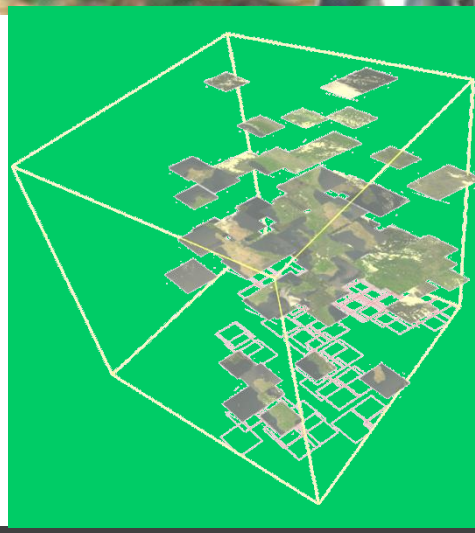
[rasdaman screenshots]



Sample Application: Database Visualization



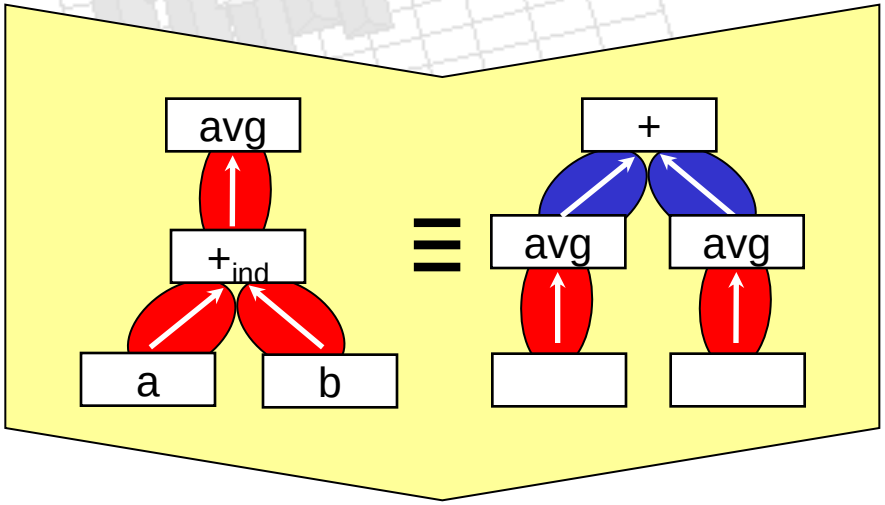
```
select
  encode(
    struct {
      red:   (char) s.image.b7[x0:x1,x0:x1],
      green: (char) s.image.b5[x0:x1,x0:x1],
      blue:  (char) s.image.b0[x0:x1,x0:x1],
      alpha: (char) scale( d.data, 20 )
    },
    "image/png"
  )
from SatImage as s, DEM as d
```



[JacobsU, Fraunhofer; data courtesy BGS, ESA]

Q'Optimization 1: Query Rewriting

```
select avg_cells( a + b )
from   a, b
```



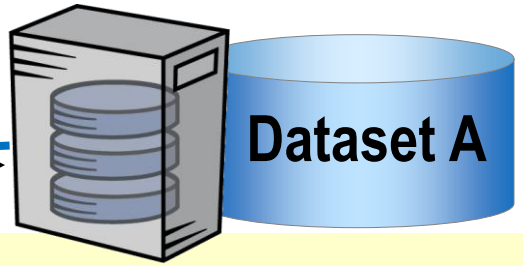
Tile stream
high traffic

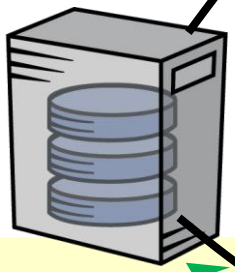
Scalar stream
low traffic

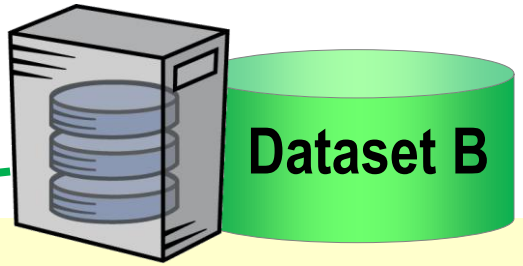
```
select avg_cells( a )
     + avg_cells( b )
from   a, b
```

- *understood:*
  heuristic optimization
  – 150 rules in rasdaman [Ritsch 2002]
- *partially understood:*
  cost-based optimization

Q'Optimization 2: Federated Array Processing







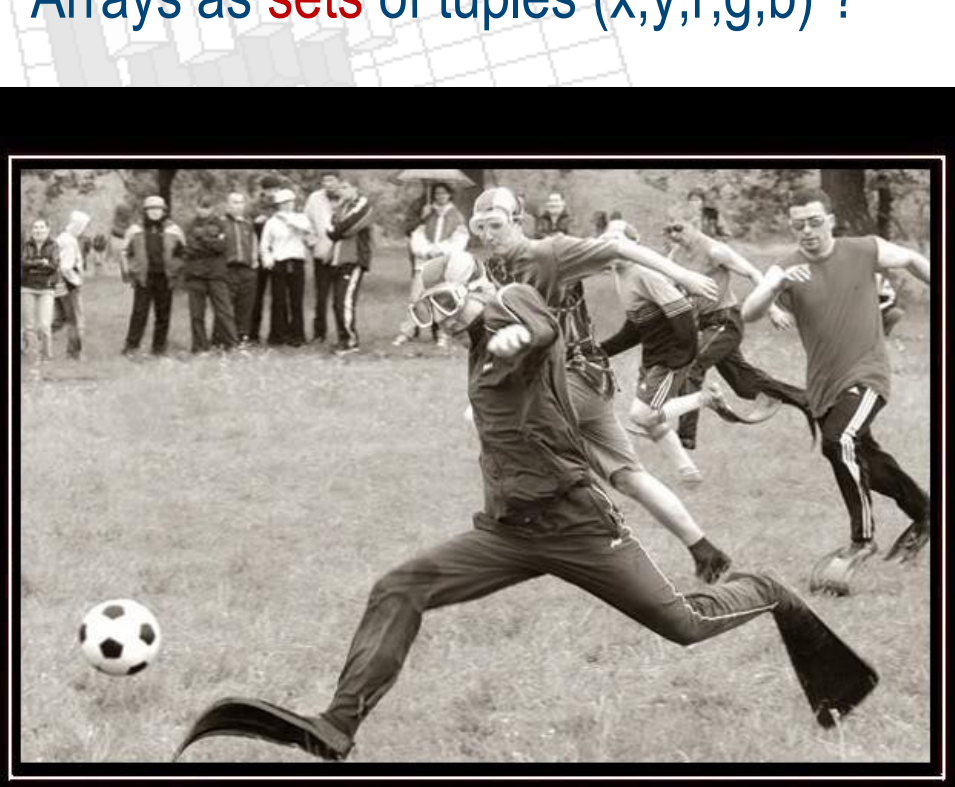

```
select
    max((A.nir - A.red) / (A.nir +A.red))
from A
```

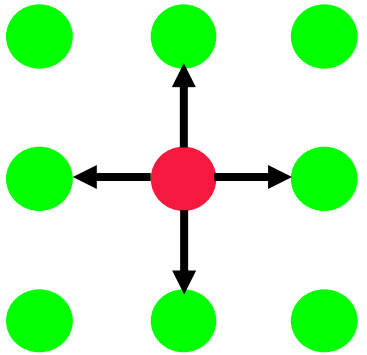
```
select
    max((A.nir - A.red) / (A.nir + A.red))
  - max((B.nir - B.red) / (B.nir + B.red))
from A, B
```

```
select
    max((B.nir - B.red) / (B.nir + B.red))
from B
```

One Size Fits All?

- Distinguishing property of arrays: n-D Euclidean neighborhood
- Arrays as sets of tuples (x,y,r,g,b) ?



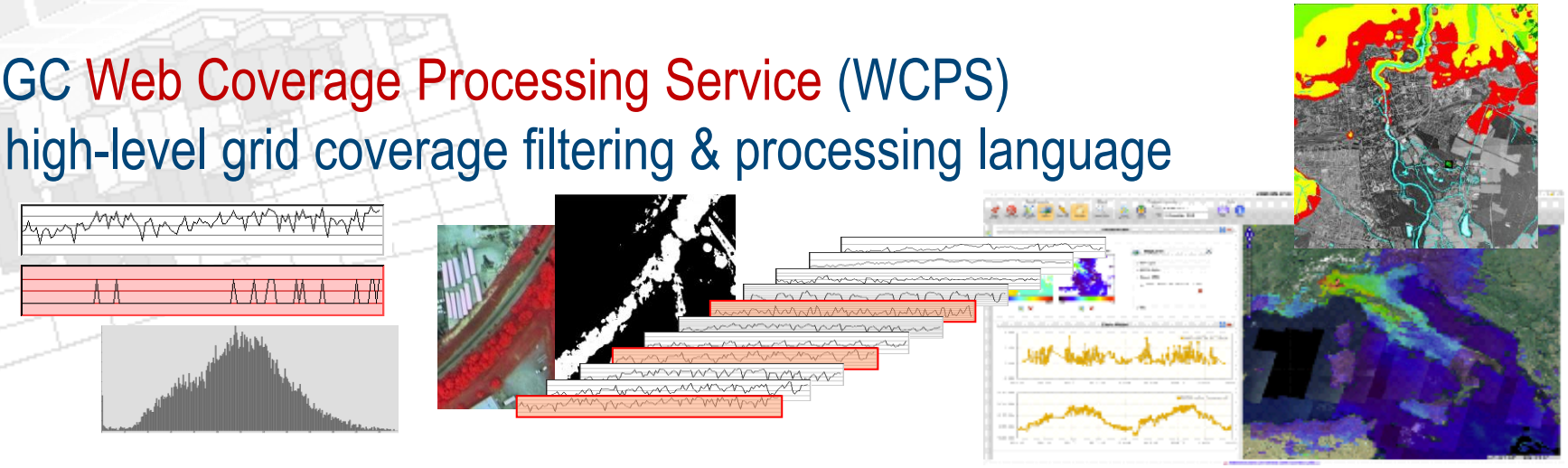


Use Case 2: Sat Imagery Standards

- Open Geospatial Consortium (OGC): geo data & service standards
- Coverage = space/time varying phenomenon [ISO,OGC]
  - n-D raster data, point clouds, meshes
  - Conceptually: GML, in practice: GeoTIFF, NetCDF, ...
  - Like metadata!
- Web Coverage Processing Service (WCPS): coverage QL
  - Integrated data/metadata queries
    → integration WCPS + XQuery
    → WCPS 2.0 draft

WCPS: the OGC Coverage Query Language

- OGC Web Coverage Processing Service (WCPS)
  = high-level grid coverage filtering & processing language



- "From MODIS scenes M1, M2, M3: difference between red & nir, as TIFF"
  - ...but only those where nir exceeds 127 somewhere

```
for $c in ( M1, M2, M3 )
where
    some( $c.nir > 127 )
return
    encode(
        $c.red - $c.nir,
        "image/tiff"
    )
```

→ $(\text{tiff}_A, \text{tiff}_C)$

WCPS + XQuery

- Ex1: „difference of red, nir bands for all coverages on Austria“

```
for $c in doc("http://acme.com")//coverage
where
    some( $c.nir > 127 ) and metadata/@region = "Austria"
return
    encode( $c.red - $c.nir, "image/tiff" )
```

- Ex2: „name & location of coverages showing some phenomenon“

```
for $c in doc("WCPS")//coverage/[ some( $c.nir > $c.red )]
return
    <id> { $c/@id } </id>
    <area> { $c/boundedBy } </area>
```

- WCPS 2.0, in progress
  - Implementation: federation of eXist + rasdaman

Semantic-Rich Interfaces

- WCPS: semantics in parseable query

```
for $c in ( M1, M2, M3 )
return encode abs( $c.red - $c.nir ), "hdf" )
```

- WPS: semantics in human-readable text

```
<ProcessDescriptions ...>
  <ProcessDescription processVersion="2" storeSupported="true" statusSupported="false">
    <ows:Identifier>Buffer</ows:Identifier>
    <ows:Title>Create a buffer around a polygon.</ows:Title>
    <ows:Abstract>Create a buffer around a single polygon. Accepts the polygon as GML and
provides GML output for the buffered feature. </ows:Abstract>
    <ows:Metadata xlink:title="spatial" />
    <ows:Metadata xlink:title="geometry" />
    <ows:Metadata xlink:title="buffer" />
    <ows:Metadata xlink:title="GML" />
    <DataInputs>
      <Input>
        <ows:Identifier>InputPolygon</ows:Identifier>
        <ows:Title>Polygon to be buffered</ows:Title>
        <ows:Abstract>URI to a set of GML that describes the polygon.</ows:Abstract>
        <ComplexData defaultFormat="text/XML" defaultEncoding="base64" defaultSchema="http
://foo.bar/gml/3.1.0/polygon.xsd">
          <SupportedComplexData>
                                                                    1,1           Top
```

Summary

- Big Data Variety includes variety of data types
  - Sets/bags, graphs, arrays, documents, ...
- Each data type calls for its specific operations
  - Domain-Specific Languages (DSLs)
  - Likely: data management subsystems
- The Future (IMHO):
  Dedicated data languages + next-gen mediators
  - Cross-model integration & optimization & parallelization
  - First, local approaches known (XQuery+WCPS; SQL+arrays)
    ...general solution?